MATT Product Documentation

Joel Henry, University of Montana – 2/2005

What it is: MATT (Matlab Automated Test Tool) is a set of software tools that automate testing of Matlab/Simulink models and automatically generated source code. MATT allows the user to quickly specify test data for the inports, specify the test step duration and total test time, define defect criteria for outports, execute a simulation, and investigate the output values for each outport manually and through graphs. MATT can execute these same tests on automatically generated source code. Two supporting tools, GIST (Graphical Input Specification Tool) and RATT (Reliability Analysis Test Tool) allow graphical specification of input values for inports and outports.

Features: MATT allows you to:

1. select the entire Simulink model, or any subsystem for testing;
2. specify the time step and test duration;
3. quickly generate the input values for each inport for each time step using built in functions, user specified functions, and graphical tools;
4. specify defect criteria for each outport or combination of outports;
5. quickly execute tests using simulation or test the automatically generated code;
6. detect defects based on outport defect criteria; and
7. configure and execute tests on a system or subsystem based on combinations of inputs.

Benefits: MATT allows developers and testers to quickly and easily configure and execute tests on Matlab/Simulink models. These tests can be stored and re-executed when the models change and source code is regenerated. Defects can be quickly detected and evaluated. A development team could use MATT for unit and integration testing as a seamless portion of the development process. A test team can use MATT to test Matlab models and automatically generated source code.

Successes: SAIC has used MATT on the STEREO project. MATT was able to test almost all portions of the STEREO model (some third-party blocks presented problems for MATT). MATT is easy to install, has a short learning curve, and is easily configured. A large number of tests can be configured, executed, and investigated in a short time. RATT allows test results to be combined to identify shortfalls in testing coverage.

Contexts in which it is best used: MATT is best used by those who are familiar with Matlab/ Simulink. Further, MATT needs to be placed in the development environment where model developers can use MATT consistently. MATT should also be used by testers familiar with Matlab/Simulink and control systems.

Compare with alternative products or technologies. The Software Productivity Consortium offers a TVEC (www.t-vec.com), a tool that generates test scripts, executes tests, and analyzes results of Matlab/Simulink models. This tool uses the stored Matlab/Simulink model. It is powerful and expensive and has only been available when an organization commits to a full evaluation project. It is not generally available for review in any other form (limited functionality or time). However, TVEC maintenance is commercially supported while MATT relies on continued external funding for maintenance. Reactive Systems also offers Reactis, a Simulink test tool. This tool automatically generates test cases and data from the Simulink and Stateflow models. The tests are then run automatically as well. This is a powerful tool with functionality that overlaps MATT. However, Reactis automatically generates test cases that the user can then change as needed. This may or may not be more powerful than MATT (this is user dependent – some users want to specify tests, others value a tool's ability to generate tests).

What will a successful collaboration look like?

A. **What will the technology provider do?** The Real-time System Laboratory at The University of Montana will provide on-site training and support upon project launch. After project launch, RTSL staff, including Dr. Joel Henry, will provide phone, email, and teleconference support, maintenance activities on MATT, and documentation as needed to support use, understanding, and suggestions on using MATT.

B. **What should the development team do?** Prior to collaboration, the development team should arrange installation and simple troubleshooting of MATT (i.e. installation, initial execution, shutdown, etc.) to insure MATT is compatable with operating system, networking, and security at the development site. Next, the collaborating team should allocate one-half day for training of those already familiar with Matlab/Simulink, or a full day for those unfamiliar with Matlab/Simulink. Finally, the development team should commit to using MATT during development and testing of Matlab/Simulink models, to providing feedback to the RTSL, and to suggesting future functionality.

C. **How will the technology provider work together with the development team to ensure a successful collaboration?** Dr. Henry has extensive consulting experience and will put this to use by focusing on development team needs, working to understand tool and methodology shortcomings, quickly providing solutions where possible, and striving to make tangible contributions to the development team goals without regard to testing tool implications.